1. Practical Number, Title, Date & Time

Practical Number – 01

Title - Introduction to Java Programming

Date & Time – 14/03/2025 (10.30 AM -12.30 PM )

1. Objectives

To understand the basic structure of a Java program, including:

* Importing Packages
* Printing Outputs
* Working with Classes, Objects, methods, & Arguments.

1. Theory/ Concepts

**Class Declaration**: In Java, all code is contained within a class. A class is a blueprint for

Creating objects. Basic syntax for declaring a class:

**class *class\_name***

**{**

***//class body***

**}**

**\*\*** Class name should start with a capital letter \*\*

**Package Importing:**  ***import java.io.\**; -** This statement imports all classes and interfaces from the java.io package.

**main Method:** The Java main method is the entry point for executing a Java program. The main method can contain code to execute or call other methods, and it can be placed in any class that is part of a program. The syntax of the main method is always:

**public static void main ( String[] args)**

**{**

***// code***

**}**

* **public:** Makes the main method accessible from outside the class.
* **static:** Allows the main method to be called without creating an object of the class.
* **void:** Indicates that the main method does not return any value.
* **String[ ] args**: This array holds the command-line arguments as strings. The first argument is stored at args[0], the second at args[1], and so on.

\*\*You can change only the name of the ***String*** array argument. For example, you can change ***args*** to ***mystringargs*** like as follow:

**public static void main ( String[] mystringargs)**

**{**

***// code***

**}**

**Object Instantiation:** Creating an object of a class is called **instantiation**. It uses the new keyword:

**ClassName obj = new ClassName();**

**Printing Outputs:** We use ***System.out.print()*** and ***System.out.println()*** for displaying outputs.

* ***System.out.print()*** - Display all outputs in same line
* ***System.out.println() –*** Display outputs separately in new lines

1. Algorithm/ Flowchart

Exercise 1: Java Basic Structure

* + - 1. Start
      2. Declare a public class named MyFirstJavaProgram.
      3. Define the main method.
      4. Print “Hi”, “This is IT1214 Practical Session 1” using System.out.print().
      5. Print “This prints a new line”, “This is in a new line” using System.out.println().
      6. End.

Exercise 2: Understanding Command-Line Arguments

1. Start
2. Declare a public class named P2.
3. Define the main method.
4. Print the command-line arguments in order.
5. End.

Exercise 3: Classes and Variables

1. Start
2. Declare a class named One with variables and methods.
3. Declare a class named Two with variables and methods.
4. Define a class named App with the main method.
5. Create objects of One and Two.
6. Assign values and call methods.
7. End.
8. Source Code

GitHub Repository:

1. Sample Inputs & Expected Output

Exercise 1

* Running “MyFirstProgram.java”

Command: java MyFirstPrgram

Expected Output:

HiThis is IT1214 Practical Session 1This prints in a new line

This is in a new line

Exercise 2

* Running “P2.java”

Command: java P2

Hello My World

Expected Output:

Hello World My

Exercise 3

* Running “P3.java”

Command: java App

Expected Output:

From class One a = 10

From class One b = 0

From class One a = 10

From class One b = 100

From class Two a = 0

From class Two c = 0

From class Two a = 20

From class Two c = 30

1. Observation

* The first program shows that **print()** keeps printing on the same line, but **println()** moves to a new line.
* The second program needs at least three inputs from the command line. If not, it gives an error called **ArrayIndexOutOfBoundsException.** That commands are printed as per the index position which demonstrates the use of the **String[ ]args** array.
* The third program uses object-oriented programming (OOP) ideas:
* Variables inside a class get default values (like 0 for numbers).
* After changing the variables, the new values are displayed.

1. Conclusion

In this session, I:

* Practiced how to print output using **print()** and **println().**
* Learned how to use command-line inputs properly.
* Studied the basics of object-oriented programming with classes and objects.
* Understood how variables behave and how they get default values in Java.